RELEASE NOTES FOR MARI 2.5V1

Release Date

07 October 2013

Best Bits of 2.5v1

- Mari support for Mac OS X 10.8.4 Mountain Lion, or above, with NVIDIA graphics cards, and over 150 general improvements and optimizations.
- A Custom Shader API, and accompanying API documentation, has been created. Documentation can be accessed from Help > SDK > Custom Shader API.
- The shader code has been re-factored to improve usability and to adhere to the coding standards.
- The smart selection tool performance has been significantly improved and now has a progress bar, as well as the ability to be a cancelable action.
- Key modifier behavior in the layers view has been updated so that holding down different modifier keys while dragging layers allows you to switch between copy, share, or move modes more easily.
- The ability to extract a snapshot in the Snapshots palette has been re-enabled.

System Requirements

Officially Supported Operating Systems

- Mac OS X 10.8.4 (Mountain Lion), or above
- Windows 7 (64-bit), or above
- Linux 64-bit operating system (Fedora 12 or Red Hat Enterprise Linux 5.4)

NOTE: Some newer Linux distributions have changed their network device configuration handling and won't work with Mari without some configuration changes. Known affected distributions at this time include Fedora 16, Red Hat Enterprise Linux 6, and CentOS 6 on certain, notably Dell, hardware. Please see our FAQs for more information: http://www.thefoundry.co.uk/support/faqs/#question_97

Minimum Hardware Requirements

- Quad-core processor
- 250GB disk space available for caching and temporary files (or a minimum of 50GB if you're working on a small project)
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD graphics card with at least 1GB of RAM and OpenGL 3.2 support (see Mac OS X Tested Graphics Cards and Windows and Linux Tested Graphics Cards below)

NOTE: Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

Virtual Memory Requirements

It is recommended that virtual memory be available for Mari to use, especially when handling large projects. The use of virtual memory improves stability, helping to prevent data loss on large projects.

Mac OS X Tested Graphics Cards

- NVIDIA GeForce GT 650M
- NVIDIA Quadro (Fermi Series) 4000

Windows and Linux Tested Graphics Cards

- NVIDIA GeForce GTX 480
- NVIDIA GeForce GTX 580
- NVIDIA GeForce GTX 680
- NVIDIA Quadro (Fermi Series) 4000
- NVIDIA Quadro (Fermi Series) 5000
- NVIDIA Quadro (Fermi Series) 6000
- NVIDIA Quadro K2000
- NVIDIA Quadro K4000
- NVIDIA Quadro K5000
- AMD FirePro W7000
- AMD FirePro W8000
- AMD FirePro W9000
- AMD Radeon HD 7850
- AMD Radeon HD 7950

Please download and install the latest graphics driver for your card from the NVIDIA or AMD websites.

NOTE: Ensure that you are using 8.982.8.1 (AMD Catalyst 12.10) drivers or higher for AMD cards.

Due to the number of cards available in the market, we are unable to test every card and, therefore, cannot officially qualify all of them individually. However, if your machine/GPU specifications meet the requirements mentioned, theoretically Mari should work fine for you. If you encounter any issues, please contact support@thefoundry.co.uk.

New Features

Mac Support

- Mari now supports Mac OS X 10.8.4 Mountain Lion with NVIDIA graphics cards. See Mac OS X Tested Graphics Cards for more information.

Graphics Cards

- The following NVIDIA cards have been added to the Windows and Linux tested list:
  - Quadro K2000
  - Quadro K4000
  - Quadro K5000
- Support for the NVIDIA Texture barrier extension **GL_NV_TEXTURE_BARRIER** has been added. If this is present, it helps reduce performance slowdown while painting.

Preferences

Windows and Linux: Support for VSync options have been added. A new preference control can be found in the **Mari Preferences** dialog, under **GPU** > **General** > **VSync**.

Set the dropdown preference to either

- **On**,
- **Off**, or
- **System**.

Setting the option to **On** forces the canvas to update at the screen refresh rate. **System** lets the OS determine the behavior, which could be on or off. Turning it **Off** allows the canvas to update at any time, but can result in

tearing if the canvas is running faster than the monitor's refresh rate. This setting is OS- and driver-dependent.

Shaders

- A Custom Shader API and accompanying API documentation have been created. This allows you to create your own shaders to use alongside the default shaders in Mari. The Custom Shader API documentation has been added to the **Help** > **SDK** menu in Mari.

  The Shader API now contains Perlin and Simplex noise functions.
- The shader code has been re-factored to improve usability and to adhere to the coding standard. Aspects, such as the appearance and behavior of the code, have been made more consistent by these improvements.
- The option to set whether or not to **Perturb Normals** has been added to the **Shaders** palette in order to stop displacement changing the surface normals, tangents, and bitangents.

Miscellaneous

- The **Smart Selection** tool now has a progress bar to indicate how long is left until Mari completes the action. The action can also be canceled at any time. On the **Tools** toolbar, with the **Select** tool active, the **Smart SelType** and **Smart SelAngle** fields have been renamed to **Smart Type** and **Smart Angle**.
- BUG ID 34544 - The shortcut to step backwards for animation has been added to the **Shortcut Manager** again. The shortcut is **Ctrl**+left arrow.
- BUG ID 34666 - The blend modes **Add Normal Maps** and **Mix Normal Maps** have been added to the list of **Painting Mode** blend modes. These modes allow accurate modulation between two RGB-encoded normal map layers.
- BUG ID 35315 - You can now press the **Tab** key in the **Layers** palette to create a layer by typing in the name, as with Nuke's **Tab** key node creation.
- BUG ID 36906 - Two new adjustment layer types, **Linear To Log** and **Log To Linear**, have been added to convert from a linear colorspace to a log colorspace. The conversion is the same as that found in Nuke.
- BUG ID 37128 - A new set of Perlin and Simplex noise functions with 2D, 3D, and 4D inputs has been added.

Feature Enhancements

Ptex

- The ability to import Ptex quad faces from a layout used for bilinear interpolation has now been added.

- You can now export Ptex quad faces suitable for bilinear interpolation.
- The Ptex triangle-to-quad warping conversion methods have been improved. Tolerance checks also prevent NaNs from causing out-of-bounds accessing on image data.

Shaders

- The shader component name **Specular IOR** of the Cook Torrance shader has been renamed to **Specular IOR Weight**.
- Custom patch sizes now use the global **ShaderInputs** default patch size when no patch size is specified.
- Error handling during shader bakes has been improved.
- BUG ID 36932 - You can now manually enable or disable mip-map generation and usage on images used in shader network nodes. The use of mip-maps on spherical projector layers has been deprecated.

Layers

- BUG ID 34568 - Procedurals can now be baked and exported using the **Export for Maya** script, under **Python** > **Examples**.

  Please refer to 34568 in the Developer Notes for additional information on this enhancement pertaining to developers.
- BUG ID 34818 - **Share As Channel** in the **Layers** menu and **Layers** palette context menu has now been renamed **Share Layers As Channels**, and allows you to share all selected layers into a new channel.
- BUG ID 34879 - Blending properties could not be changed on layers that had been cached.

  Blending properties are still disabled for layers cached using the **Cache Up To Here** function.
- BUG ID 36636 - The controls used to adjust the **X**, **Y**, and **Z Offset** values for the **Sphere Map Projector** and **Cube Map Projector** procedural layers now include incrementers. This enables you to adjust the values with either the keyboard or the mouse. You can also hold **Ctrl**/**Cmd** while adjusting the values to adjust the incrementer range.
- BUG ID 36896 - New mask image sets and paintable layers added to mask stacks now have their colorspace set to **Scalar** so that they are not processed through the Color Manager.
- BUG ID 38080 - A new **Components** dropdown has been added to the **Constant** procedural layer. **Components** allows you to choose whether the **Constant** procedural applies to **RGB** or **RGBA** components.

Shorctus and Preferences

- The preference **Max Batch Index Size** in the GPU settings has been removed.
- BUG ID 34810 - In the **Manage Keyboard Shortcuts** dialog, the search function has been improved so that incomplete search parameters still return results that contain part of the search string.
- BUG ID 36216 - The shortcut **Alt**+drag and drop for channels now duplicates the dropped channel.
- BUG ID 37699 - Mac OS X only: Tools that used the **Meta** key have now been switched to use **Cmd**+**Shift** instead. The **Meta** key (**Ctrl**) on the Mac operating system is used to bring up the context menu.

Miscellaneous Improvements

- The canvas maximum frame rate has been improved by changing the update mechanism.
- The memory usage in selection topology for the **Smart Select** tool has been optimized.
- Image format extension checks are no longer case-sensitive for many nodes in Mari.
- Intensity values have been added to light sources to allow scaling for diffuse and specular colors. Intensity sliders are now included in the **Lights** palette for adjusting these values.
- BUG ID 12308 - The **-V** (upper-case) command line option has been added in addition to the existing **--verbose** and **-v** flags to show the log in the console.
- BUG ID 34263 - The example script **Export Channels** under **Python** > **Examples** has been renamed to **Export Image Sets**, and has been updated to support layers.
- BUG ID 37032 - The **Color Space** color profile can now be edited. Extra filters can be added and removed, and existing filters can be re-ordered.

Bug Fixes

- Ptex: Quad warping sometimes occurred when image data was set to an option other than **Byte**.
- In the **Ortho/UV** view, paint appeared incorrectly in both views.
- Procedurals that were converted to paintables did not have correct pixels around patch geometry edges.
- Converting nodes occasionally deleted their VersionUUID.
- Custom nodes were problematic and have been deprecated.
- It was possible to create invalid image set patch sizes.

- Mari occasionally crashed when failing to create a layer shader.
- Ptex models with concave quads resulted in quad warping during rendering.
- The use of **gl_PrimitiveID** in the **GLColorSwatch** caused performance issues. The use of this has been removed.
- BUG ID 13435 - The grid and lights were sometimes clipped in **Ortho** view.
- BUG ID 27657 - Linux only: Mari did not always correctly restore the window position and size.
- BUG ID 32160 - Unbaked paint that was used for creating bump maps changed its bump effect when zooming in or out.
- BUG ID 32279 - Mari didn't remember its maximized state between sessions, and opened in the default state every time.
- BUG ID 33046 - The **Tool Properties** toolbar was incorrectly disabled when the default layout, or a saved layout, was loaded.
- BUG ID 33322 - On systems set to the Russian locale, exporting to **.psd** occasionally failed with the error "Error writing data to PSD file: Pstring too long, must be <= 255 chars."
- BUG ID 33360 - Mari became unresponsive when using the **Smart Selection** tool in **Face** mode.
- BUG ID 33609 - Default lighting values in shaders did not match those used in pre-2.0v1 versions of Mari.
- BUG ID 34414 - Windows only: When using a mouse, the cursor did not update over the canvas when the right-click menu was open.
- BUG ID 34444 - Renaming a channel did not update the name in all instances where the channel name occurred in Mari.
- BUG ID 34594 - The default base layer on new objects was not RGBA.
- BUG ID 34674 - Windows only: The **Channel Presets** dialog did not generate the correct file import templates when scanning directories.
- BUG ID 34804 - Windows only: An infrequent issue on AMD graphics card drivers caused shader baking actions, such as flattening, merging, and caching, to swap around parts of the texture.
- BUG ID 34892 - When masking was disabled, paint baking projected paint through to the other side, even if **Projection** was set to **Front**.
- BUG ID 34955 - Linux only: Importing a **.psd** file using the background importer caused a crash.
- BUG ID 35030/36945 - Patch resolutions did not save correctly and were lost between Mari sessions.

  You can no longer set different patch sizes for different animation frames. All animation frames must keep the same patch resolution.

- BUG ID 35050 – Pressing the modifier keys while the mouse button or tablet pen were held down did not engage navigation.
- BUG ID 35058 – Mari did not update the copy, move, and share modes of layer dragging when you changed the modifier key.
- BUG ID 35061 – Mari did not calculate values over 1.0 in the **Screen** blend mode.

  The blend mode is now calculated so that it is in line with the standard that Nuke uses.
- BUG ID 35092 – The option to **Export Everything** displayed a dialog for each channel/object exported.

  This now only happens if the $ENTITY part of the template is missing.
- BUG ID 35100 – Updating a specular or diffuse shader did not update the shader GUI.
- BUG ID 35160 – The **Height As Normal** layer caused a render error when it was created above a layer cached with **Cache Up to Here**.
- BUG ID 35218 – **Channel Presets** always resized the images to fit the patches, regardless of the setting in the **Channel Presets** dialog.
- BUG ID 35237 – **Import into New Channel** did not correctly set the default resolution for the new channel when the **Resize patches to fit images** option was selected.
- BUG ID 35242 – **Share Layers as Channels** did not correctly set the default image resolution for the new channel. The resolution information in the **UV** view also did not show the resolution of the patch of the current channel.
- BUG ID 35275 – Converting projects containing cube maps caused Mari to crash.
- BUG ID 35290 – The **Bicubic** image filtering option caused a shader compilation error.
- BUG ID 35326 – The patch size of additional patches in new versions of objects did not match the channel size.

  Mari now attempts to guess the resolution of the missing patch when a new version has been added.
- BUG ID 35338 – Using the **Alt**+drag and drop shortcut on layers occasionally caused Mari to crash.
- BUG ID 35366 – There were calculation errors in applying a **Hue Shift** adjustment layer to HDR textures.
- BUG ID 35382 – Caching a layer sometimes caused a rendering error.
- BUG ID 35391 – Removing the last layer from a layer stack did not remove the layer parameters from the UI.
- BUG ID 35464 – The **Project** functionality in projectors sometimes missed areas when using small images.

- BUG ID 35501/36894 - The **clamp()** function was incorrectly accessible from various blend modes and the HSV adjustment layer.
- BUG ID 35650 - Object Space Linear Gradient was not operating for displacement for deferred rendering.

  Additional information related to this bug has been included for developers in the Developer Notes.
- BUG ID 35728 - Undo stopped working after using the **Paint Buffer Transform** tool.
- BUG ID 35745 - Flattening or merging layers caused Mari to crash.
- BUG ID 35751 - The **Lighting** option was missing from the **Quick Project** and **Quick Unproject** settings.

  The pseudo lighting layer in the **.psd** layer output of **Quick Unprojection** is now generated in the **Multiply** blend mode.
- BUG ID 35765 - Disabling visibility of a layer sometimes caused Mari to also disable color management.
- BUG ID 35774 - The use of shadows stopped the canvas from drawing in certain modes.
- BUG ID 36344 - It was not possible to stamp paint through to an image using the ' (apostrophe) hotkey on Japanese keyboards.
- BUG ID 36445 - When flattening a channel, the resolutions of custom patches were being ignored and reset to the default size.
- BUG ID 36482 - An object without the "1001" UDIM patch was not loaded correctly after saving its project, resulting in failures when exporting channels.
- BUG ID 36509 - Hiding a patch or its parent object caused Mari to fill the patch.
- BUG ID 36612 - Exporting flattened, non-current objects using Python or the **Export Everything Flattened** method exported the textures of non-current objects incorrectly.
- BUG ID 36672 - Mari sometimes crashed when changing the **LUT File** in the **Color Manager**'s **Color Space** profile if the **Gain** and **Gamma** values differed.
- BUG ID 36736 - The index search mode of **Select Range** selected incorrect patches.
- BUG ID 36749 - The OpenColorIO **extrapolate** option was available when no LUT had been set.
- BUG ID 36839 - It was possible to create a Ptex image with a resolution that did not match existing image sets in the geometry.
- BUG ID 36866 - Using **Export Flattened** exported channels at a lower resolution than the channel resolution when the channel contained a cached layer created at the lower resolution.

- BUG ID 36909 - It was possible to clamp the **Levels** adjustment layer.
- BUG ID 36952 - Reloading files containing multi-object projects used incorrect channels on the shader.
- BUG ID 36972 - The **Zoom Paint Buffer** mouse cursor did not appear over the canvas.
- BUG ID 36998 - Cached nodes were incorrectly triggering node upgrades.
- BUG ID 37028 - Copying and pasting a channel that contained a shared paintable layer caused the resulting channel to contain empty layers.
- BUG ID 37032- When moving filters with the up or down arrow buttons, the **Color Manager**'s filter collection view sometimes destroyed the filters.
- BUG ID 37057 - The **Black Input Level** option on the **Levels** adjustment layer introduced color value artifacts.
- BUG ID 37097/37288 - When using the tow brush, shader baking (such as layer caching, flattening, and grouping layers) resulted in patches not bleeding correctly to expose UV seams.
- BUG ID 37183 - **Through** projection did not work when the **Mask Disabled** option was checked.
- BUG ID 37232 - Linux only: Navigation did not work as expected when panning and rotating with a Wacom tablet.
- BUG ID 37296 - Rotating the paint buffer using the **Transform Paint Buffer** tool resulted in subsequent paint strokes not following the new transform correctly.
- BUG ID 37466 - Selecting faces within a model that had degenerate faces caused Mari to crash.
- BUG ID 37521 - Exporting unflattened channels using the threaded channel exporter with partial UDIM lists caused Mari to crash.
- BUG ID 37606 - The **Paint Through** image shifted when the filtering mode was changed to **bicubic**.
- BUG ID 37615 - The cursor image did not change to represent size adjustments when using the **Paint Through** tool.
- BUG ID 37629 - Painting with the Intuos 3 tablet produced a fully opaque paint splat when the pen was first pressed down.
- BUG ID 37641 - The **line width** control in the **Vector Inspector** tool has been deprecated.
- BUG ID 37694 - The **Blur** filter previews caused the canvas to stop rendering.
- BUG ID 37695 - It was not possible to move lights or objects around the scene with the **Pan Object** tool.

- BUG ID 37696 - Spheres that represented lights were rendered in front of scene objects instead of in depth order.
- BUG ID 37726 - Opening **Levels** or **Clamp** filter dialogs sometimes caused Mari to crash.
- BUG ID 37736 - Mari sometimes displayed the **Converting Layers** dialog on opening a project that was created in the same version.
- BUG ID 37740 - The **Add Noise** filter stopped the canvas rendering correctly.
- BUG ID 37743 - The use of layer color tags caused the node upgrade system to attempt to upgrade the project on load as well.
- BUG ID 37759 - The **Wireframe** fill option under **Patches** > **Fill** did not work correctly.
- BUG ID 37789 - Wireframe rendering did not draw correctly in **Perspective** view.
- BUG ID 37825 - Adding to an existing selection caused the **Smart Selection** tool to deactivate.
- BUG ID 37827 - Clicking the color swatch in the HSV color picker displayed multiple color pickers.
- BUG ID 37911 - Saving a project after converting a procedural to be paintable caused Mari to crash.
- BUG ID 37920 - Copying and pasting layers sometimes caused Mari to become unresponsive, produce rendering errors, or create strange layer stack results.
- BUG ID 37946 - Right-clicking on a non-current cached layer caused the context menu to redraw a number of times.
- BUG ID 37995 - Only the selected object showed a wireframe rendering, when enabled, instead of all objects in a scene.
- BUG ID 38046 - The ability to use any image imported into the **Image Manager** as part of a multi-layered **.psd** file was working incorrectly so that painting with the **Paint Through** tool only let you select one layer.
- BUG ID 38084 - Using **Alt**+**Tab** to select a different window during baking corrupted the bake.
- BUG ID 38132 - Collapsing or expanding groups deleted the cache from the group and any sub-layers.
- BUG ID 38176 - When one light was modified or selected in the **Lighting** palette the lighting was sometimes shown on the canvas in the **UV** view.
- BUG ID 38194 - Using the **Clone** tool in the Ortho and UV views and moving the cursor between the two, corrupted a working bake operation.
- BUG ID 38210 - Copying masks from one channel to another caused Mari to crash.

Known Issues & Workarounds

Mari Tools

- BUG ID 15810 – The black borders at the edge of the canvas in **Perspective** view are selected, if an object overlaps the borders when a selection is made using the **Marquee Select** tool, with the selection mode set to **Magic Wand**.
- BUG ID 13640 – The **Blur** tool can be slow to use on the initial stroke.

  Wait for Mari to process the blur before applying a second stroke.
- BUG ID 13481 – Making a selection using the **Marquee Select Tool** with the selection mode set to **Magic Wand** sometimes ignores regions that are completely surrounded by other selected regions, despite differences in color.
- BUG ID 13394 – Using the **Select Items** tool with the **Facing** set to **Front** to select and hide a portion of faces causes some of the faces within the selection to remain visible when zoomed in.

  To catch all selected faces, either:

  • select **Facing > Through** instead of **Front**, or

  • zoom in closer to the object.

Shaders

- BUG ID 34729 – Mari displays a rendering error on the canvas when it is unable to create a shader. More information has been included to help you determine the cause of the error. Some solutions might be to hide groups and layers, or to cache parts of your layer stack until a shader can be created.
- BUG ID 34679 – On extremely large projects, issues can arise with shader limits and reaching the maximum allowed texture slots available. To avoid reaching these shader limits on large projects, try the following workarounds:

  • hide groups and layers, or

  • cache groups and layers.

Layers

- BUG ID 34690 – Flattening or caching layers or channels on complex projects may cause Windows to reset the graphics driver due to the long processing time. To work around this issue, you can try to flatten or cache fewer layers at a time, or reduce the value of the **Max Render Size For Baking** setting. This setting can be found under **Preferences** > **GPU** > **Baking and Projection**.

  Reducing this size breaks the flattening or caching operation up into smaller pieces, which individually take less time to calculate, and thereby avoids a Windows graphics driver reset.

- BUG ID 26460 – Painting a mask in a Mask Layer Group sometimes results in unexpected paint results. To prevent this from happening, either:
  - Use a white "color" layer at the bottom of your mask stack. Any layer used over this initial "color" layer should then be fine, or
  - If you want to create a mask in a Mask Layer Group, simply add another layer on your Mask Layer Group instead, and paint white into it to create a mask.

Importing and Exporting

- BUG ID 29386 – When using the **Export for Maya** script, Maya's viewport may incorrectly show some patches as transparent. This can be resolved by selecting **High Quality Rendering** or **Viewport 2.0** from the Renderer menu within Maya.
- BUG ID 16324 – Windows only: you cannot currently import an image into a channel using a relative file path. To work around this, use an absolute path when importing images.
- BUG ID 14985 – There may be a slight pause after importing textures when creating new projects while Mari saves the project.

Nuke<>Mari Bridge

- BUG ID 23010 – Nuke<>Mari Bridge: If Mari crashes when receiving incoming components from Nuke when the **Virtual Texture Type** is set to **Float**, lower the **Virtual Texture Size** to a value below 8192x8192.
- BUG ID 19829 – Nuke<>Mari Bridge: Unprojecting a displaced piece of geometry does not project correctly in Nuke.
- BUG ID 19780 – Nuke<>Mari Bridge: A projector created in **Ortho** view in Mari does not re-project correctly in Nuke.

Python

- BUG ID 37601 – Python: The **QFileSystemModel** class PythonQt binding is missing. The **QDirModel** class may be used in some cases as a workaround.
- BUG ID 16616 – Python: PythonQt sometimes crashes when adding temporary objects to layouts, or compound widgets such as QTreeWidget. This is because of incorrect reference counting. To work around the problem, always store a reference to every GUI item until you are sure it is no longer needed.

Ptex

- BUG ID 17626 - It can take a long time to import very large or very high polygon count Ptex models.

  The workaround is to assign a small uniform face size (1x1 or 2x2) on import, and then increase the resolution of the relevant bits of the model as necessary after loading.
- BUG ID 17618 - Ptex does not bake properly if the resolution of the face is too small.

  The workaround is to increase the resolution of the selected faces you are having problems with.

Graphics Cards

- BUG ID 37378 - Windows: With driver version 320.49 and before, we've seen performance issues as well as timeout detection and recovery issues with the NVIDIA GeForce GTX Titan graphics card.

  The beta driver 326.80 appears to fix these problems. Please try to update the graphics card driver version if you are experiencing these issues.
- BUG ID 36260 - The AMD Catalyst 13.4 driver has been blacklisted due to a known issue resulting in baking artifacts.

  If you experience any problems of this nature, try rolling back to the 13.1 driver, as this should resolve the issue.
- BUG ID 18457 - Using NVIDIA graphics cards from the **Fermi** series with drivers older than version 270 results in various rendering issues when the **Virtual Texture Type** is set to **Half** or **Float**.

  To resolve this, please download and install the latest graphics driver for your card from the NVIDIA website.
- BUG ID 12567 - Enabling **Sync to VBlank** in NVIDIA settings can drastically reduce Mari's performance. If you experience very slow interaction, even with low-polygon models, on one of the Windows and Linux Tested Graphics Cards, navigate to:
  - Linux: **NVIDIA X Server Settings > X Screen 0 > OpenGL Settings** and turn off **Sync to VBlank**.
  - Windows: **NVIDIA Control Panel > 3D Settings > Manage 3DSettings > Vertical Sync > Force off**

  Then, restart Mari.

Miscellaneous

- BUG ID 33293 - Linux: Launching Mari with the language set to one without certain character symbols resulted in Mari failing with an error that the specified transform could not be loaded.

To work around this, set the locale (language) to English.

- BUG ID 31946 - Sometimes paint is not baked because of memory management issues on the graphics card. This can be due to issues such as a high resolution paint buffer, a high bit-depth paint buffer, large virtual texture size, or even a large scale value on the paint buffer transform. These issues can usually be identified by **glError: 0x505 out of memory** messages in the log.

  Try reducing any or all of these values to prevent it happening. Graphics drivers are continually improving, so it's also worth checking whether upgrading your drivers resolves the problem.
- BUG ID 20510 - If you find that the startup time for Mari is longer than usual, please check that the LIC files in your RLM licensing data folder do not refer to obsolete server ports. If they do, place them in another directory and restart Mari.
- BUG ID 20021 - Textures in the canvas intermittently switch between lower and higher resolutions.

  This issue is more likely to occur if your virtual texture resolution is low, and you're working on a complex model with displacement. Possible work-arounds include increasing your virtual texture size, reducing the number of channels Mari has to access at once (for example, by reducing the number of channels required for the current shader), to reduce the patch resolution of patches in the channels used in the shader, or to use a smaller canvas window or monitor.
- BUG ID 14201 - Linux only: Mari becomes unresponsive after the system is woken from sleep.
- BUG ID 13700 - Adjusting the **Camera > Perspective** settings for a **Projector** is not reflected on the canvas until the **Projector** is made Current.
- BUG ID 13571 - Launching a new version of Mari for the first time, when a config file exists from a previous version, sometimes results in an object not appearing in the Ortho view.

  To solve this, close Mari, delete the following config file and relaunch Mari:

  - Linux: **~/.config/TheFoundry/Mari.conf**
  - Windows: **C:/Users/<login>/.mari/TheFoundry/Mari.ini**
- BUG ID 13294 - Windows: Mari sometimes crashes when trying to load data on large projects due to the program exhausting all Window manager objects.

  To reconfigure the user object limit:

  - Open regedit and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\Windows**, and
  - Edit **USERProcessHandleQuota** to a larger number.

If this number gets too large, you may also have to modify **GDIProcessHandleQuota.**

- BUG ID 12102 - Current brush settings do not get saved as part of the project. Instead, Mari reverts to the default settings when you close and relaunch it.
- BUG ID 11874 - Mari doesn't recognize 3-digit padded **.obj** sequences as animation.

Developer Notes

These are the changes relevant to developers.

New Features

- The module **mari.current** has been added to the Python API.
- A new section for workflow information has been added to the Custom Shader API documentation.
- Information exclusive to shaders has been added to global uniforms and state members in the Custom Shader API documentation.
- Additional information on the **Layer.hasMask()** function has been added to the Python API documentation.
- BUG ID 34568 - The function **Layer.groupStack()** has been added to the Python API.
- BUG ID 36142 - The **Layer.siblingSharedLayerList()** function has been added. This function returns the list of other layers that share resources with this layer.
- BUG ID 36165 - The following functions have been added to the Python API:
  - **AdjustmentLayer.primaryAdjustmentType()**
  - **AdjustmentLayer.secondaryAdjustmentType()**
  - **ProceduralLayer.proceduralType()**.

  These provide a reliable way to retrieve the types of adjustment or procedural layers, and in a form that is more consistent with other functions. The existing similar functions, such as **proceduralName()**, have now been deprecated.
- BUG ID 36251 - The **GeoEntity.createStandaloneShader()** and **GeoEntity.createDiffuseSpecularShader()** functions have been added to the Python API. The function **GeoEntity.createShader()** has been deprecated.

Feature Enhancements

- New classes have been added to the **MriShaderNetwork.pro** file.
- BUG ID 27593 - The new signals **mari.projects.aboutToArchive** and **mari.projects.archived** have been added.
- BUG ID 34698 - The new function **Layer.setName()** has been added to the Python API.
- BUG ID 34727 - The **LayerStack.exportImages()**, **LayerStack.exportImagesFlattened()**, **PaintableLayer.exportImages()**, and **ProceduralLayer.exportImages()** functions have been updated to accept an optional list of UV indices to export.
- BUG ID 34805 - The new signals **Layer.nameChanged**, **Channel.nameChanged**, and **Shader.nameChanged** have been added to the Python API.
- BUG ID 35012 - The new functions **mari.Channel.setName()** and **name()** have been added to the Python API.
- BUG ID 35031 - The new functions **mari.Shader.setName()** and **name()** have been added to the Python API.
- BUG ID 35142 - The Indices page of the Python API documentation has been reformatted to one page to make searching easier.
- BUG ID 35294 - Adjustment layer vector parameters, such as in the **Grade** adjustment, were not properly processed through Python.

  These can now be set from lists or **mari.VectorN** objects, and are returned as lists.
- BUG ID 35941 - Python API: The **Channel** and **LayerStack** sub-classes of metadata have been made so that metadata can be added to a channel.
- BUG ID 36169 - **LayerStack.adjustmentLayerTypeList()** and **LayerStack.proceduralLayerTypeList()** are now static, so they can be called through **mari.LayerStack** instead of requiring a **LayerStack** object.
- BUG ID 36213 - **Layer.cacheLayer()**, **Layer.cacheUpToHere()**, and their **uncache** counterparts have been added.
- BUG ID 36250 - Metadata can now be added to shader classes in the Python API.
- BUG ID 36252 - The **GeoEntity.shaderStandaloneTypeList()**, **GeoEntity.shaderDiffuseTypeList()**, and **GeoEntity.shaderSpecularTypeList()** functions have been made static. This allows them to be called without a specific GeoEntity object.
- BUG ID 36481 - Python API export methods now have an option to generate full patch bleed.
- BUG ID 37326 - Mari's C++ API now supports using the **Color** class with the **Metadata** class, allowing you to set and get metadata/user attributes as colors.

Bug Fixes

- Context log tracking has been disabled.
- The Python API metadata flag **METADATA_REMOVEABLE** has been deprecated due to a spelling error. The metadata flag **METADATA_REMOVABLE** should be used in future.
- Mari occasionally crashed when fewer than four pairs of numbers were given to MriGradient.
- Mari unnecessarily added Change Expression undo records, even with non-undoable actions.
- BUG ID 15491 - Python API: PythonQt bindings were missing for some enumerations and types.
- BUG ID 15922/36208 - The **ImageSet.importImages()** function did not handle $CHANNEL and $LAYER token replacements properly in file names during imports.
- BUG ID 23115 - The signals on **mari.projects** had inconsistent names. Signals with new, consistent names have been added, and the old ones have been deprecated. For example, **mari.projects.projectOpened** has been superseded by **mari.projects.opened**.

  The new signal **mari.projects.opened** has an optional extra parameter to indicate whether the opened project has just been created.
- BUG ID 27592 - The default archive patch only applied to opening archives and not to saving.

  The way the path and file name parameters are combined in **mari.utils.getSaveFileName()** has changed. Please see the Mari Python API documentation for more information.
- BUG ID 34323 - Python API: The channel signals on GeoEntity were not triggered correctly.
- BUG ID 35650 - The functions **u_MriObjectCenter and u_MriObjectRadius** were not set to correct values for the evaluation stage.
- BUG ID 35940 - The **Channel.flatten()** function did not return the resulting layer.
- BUG ID 36115 - The **Application.startingProcessing()** signal was emitted for the nested start of certain sub-tasks, causing **startingProcessing()** and **endingProcessing()** signals to be unmatched.

  The **startingProcessing()** and **endingProcessing()** signals are now only called for the outer-most start and end of the progress bar.
- BUG ID 36143 - The **LayerStack.shareLayer()** function did not create a new shared layer at the top of the current layer stack when **RefLayer** was **None**.
- BUG ID 36399 - A progress dialog appeared during Python export function calls.

  This is now suppressed to prevent it from appearing during the function.

- BUG ID 36400 – **Mari::GeoEntity::systemShader()** and **Mari::GeoEntity::SystemShaderType()** functions have been added to the Python API. These functions ensure that system shaders such as **Current Channel** can be obtained using Python.
- BUG ID 36958 – The ability to set **MriExportTemplate** and **MriImportTemplate** metadata for a project was incorrectly added.

  The ability to set the **MriExportTemplate** and **MriImportTemplate** metadata of an object has been added instead.
- BUG ID 37362 – Repeatedly clicking buttons in the shader console, such as **All** and **Reload Shader**, caused Mari to crash.
- BUG ID 37509 – The **LayerStack.groupLayers() layer_name** argument was not effective.
- BUG ID 37567 – Passing **None** in **GeoEntity.setCurrentChannel()** and **GeoEntity.setCurrentShader()** caused Mari to crash.

  A check has been added to prevent passing a **None** argument.
- BUG ID 37589 – Exporting flattened hidden objects in the Python API was exporting empty textures.
- BUG ID 38039 – Python API: Calling a **mari.python.shortcut()** function with a non-existent path caused Mari to crash.
- BUG ID 38100 – Python API: The **mari.projects.closed** signal was not emitted.